

OLPC: Are We There Yet?

An update on the One Laptop Per Child Project. SAMEER VERMA

Just the other day, I ran into someone in the parking lot, who looked at the silly-looking green laptop slung across my shoulder and said “Didn’t the founder of that project ditch and disappear?” News travels fast, no matter where it comes from, and no matter how wrong it is. I’m amused.

Of course, I am talking about the One Laptop per Child (OLPC) Project, founded by Professor Nicholas Negroponte in 2005. Conceived as a learning project at MIT back in the early 1980s (with Professor Seymour Papert), it morphed into a laptop project in Cambodia in 1999 and eventually into the current OLPC Project in 2005. OLPC currently has more than two million laptops in the hands of children in more than 25 languages in more than 40 countries.

I got involved in July 2007, when I first saw the XO laptop at OSCON (thanks to Rob Savoye) and instantly fell in love with it. I quickly signed up with the developer program and got myself a beta machine. However, unlike my short-lived affairs with other gadgets, I’ve hung onto this one for more than four years now. Along the way, I became

the de facto organizer for the OLPC San Francisco volunteer community. OLPC also has found a spot on my research agenda. I now run projects in Jamaica, India and Tuva (yes, that Tuva, of Richard Feynman fame), and I help out with others in Armenia, Madagascar, Morocco and a whole bunch of other places. Making technology work in remote places has been a welcome challenge. Making it work for children the world over has been a fulfilling experience.

Hardware

Speaking of laptops, the lean, green children’s machine is quite the icon. It gathers crowds with little effort. It brings forth the inner child in many a tough adult. Let’s take a peek into what’s under the plastic. Table 1 shows the different incarnations of the OLPC XO laptop.

The XO has many cool features. It is rugged, solar-chargeable, has a sunlight-readable screen, supports mesh networking and has a very aggressive suspend-resume cycle, but the importance of these features supersedes their coolness when deployed in the field. One of my projects is in Bhagalpur,

Table 1. XO Laptop Hardware Evolution

MACHINE	CPU	RAM	STORAGE	FORM FACTOR
XO-1	AMD Geode LX 700 (x86) at 433MHz	256MB	1024MB NAND Flash JFFS2 filesystem	Laptop (green, with membrane keyboard)
XO-1	AMD Geode LX 700 (x86) at 433 MHz	256MB	2048MB NAND Flash JFFS2 filesystem	Laptop (red, with membrane keyboard)
XO-1.5	VIA C7-M (x86) at 1GHz (variable)	1024MB	4GB microSD Ext4 filesystem	Laptop (green, with membrane keyboard)
XO-1.5HS	VIA C7-M (x86) at 1GHz (variable)	1024MB	4GB microSD Ext4 filesystem	Laptop (blue, with Netbook keyboard)
XO-1.75	Marvell Armada 610 (ARM) SoC at 1GHz (variable)	512MB	Undecided (still in testing)	Laptop
XO-3	ARM (proposed)	Undecided	Undecided	Tablet



Figure 1. Red-Green-Blue Spectrum of XOs

a rural village in northern India. Bhagmalpur is close to my heart. A part of my family lives there, and keeping up with the village helps me stay grounded in their realities. I have learned much from them (I can milk

a cow!), and I am glad to be able to give back. Bhagmalpur is beautiful, but it has its own problems. Electricity is elusive at best. There's no Wi-Fi. Internet access rarely works, and when it does, it trickles. There are no classrooms; the kids sit under a tree. This is where we see the real benefits of sunlight-readable screens, solar chargers, under-a-tree mesh-networking technology and aggressive suspend-resume cycles that conserve every fraction of a watt. All of this made possible by a combination of free and open-source software.

Internet in a Box

Another neat setup we have in Bhagmalpur is an off-line server. Remember SneakerNet from the days of tapes and floppies?

SneakerNet is a wonderful concept. You can think of it as a kind of mirroring with a very high latency. Given that about 70% of the world's population still is not on the public Internet, SneakerNet approaches become important stopgap measures. OLPC has a Fedora-based image (called the XS) that is

designed to run an off-line server at the local school, village or library. Think of it as an “Internet in a box”. Yes, it’s not the entire Internet, but storage is cheap these days. You can pack plenty of content on a 2TB drive, be it Khan Academy videos, TED Talks, books from Project Gutenberg, the Internet Archive, Wikipedia and so on. This approach acts as a stopgap, so that we can get the kids access to some information at least for now. Eventually, when the Internet gets there 24/7, this server can continue to function as a conduit to the outside.

The school server runs on x86 machines (ours in Bhagmalpur draws 8 watts) and provides a plethora of services. The design of this server revolves around Moodle, a learning management system. Some of the administrative capabilities also are plugged in to Moodle, so teachers don’t have to drop to a command line to administer the server. Other services that run on the XS are Apache, PHP and PostgreSQL (to run Moodle), ejabberd (to provide XMPP-based collaborative services), Squid, DHCP, DNS and a few others. In the Bhagmalpur setup, we also use Munin to keep an eye on server health and OpenVPN to tunnel back to San Francisco to pull logs and to push content via rsync. The server not only serves up content and courses, but it also seamlessly backs up each laptop’s datastore, pushes both OS and application updates, manages theft deterrence, network access and more.

The XS is designed to follow a curricular model, complete with courses, teachers, students, enrollment and assessment all

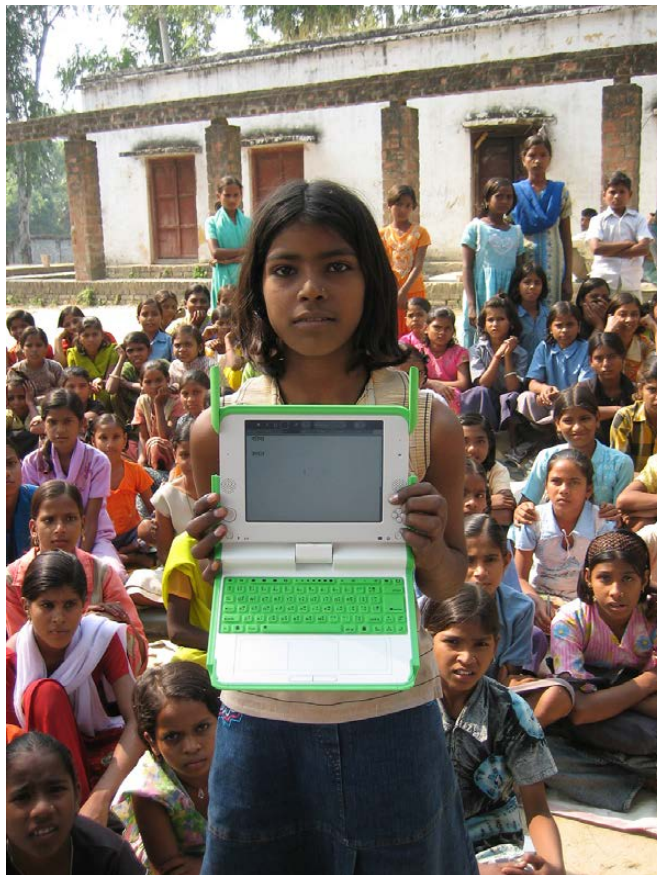


Figure 2. Garima in Bhagmalpur

managed via Moodle. However, in some instances, a curricular model isn’t suitable. Sometimes we simply don’t have a functional school in the neighborhood. In such cases, a reference library is more apt. There’s a parallel effort to fill the need for an off-line digital library, but without a specific curricular underpinning. Pathagar (in Bengali, path=lesson, agar=repository) is a Django-based digital library that serves up an XML catalog of content using the Open Publication Distribution System (OPDS). Content can be searched and downloaded using a browser or custom reader. Uruguay has been using Pathagar in production for some time now. We are testing it for Bhagmalpur (using

Debian Squeeze on a SheevaPlug) right now and hope to put it into production soon.

Deployments

OLPC projects come in all sizes. Projects that implement laptops in the field are called deployments. There are large, top-down OLPC-managed deployments and small third-party deployments managed by NGOs and volunteer communities. As Professor Negroponte puts it, uppercase OLPC is the “official” effort, whereas lowercase olpc is the community effort. Of course, one without the other wouldn’t get very far. Uppercase deployments, like Peru, are the largest with almost a million machines. Uruguay follows closely with 100% saturation at the primary school level, and Rwanda recently crossed the 100,000 laptop mark. In the lowercase olpc realm there are several projects with much less volume, but nevertheless, a strong intent. Some are in the hundreds, like my projects with the University of the West Indies in Jamaica. Others are in the tens, like the effort in Bhagmalpur. We even have some efforts with a single eBay-bought XO laptop lighting up someone’s world somewhere. It’s a spark that holds the potential to ignite.

Keeping track of smaller deployments is difficult. We have two planet feeds to keep track of various deployments (see Resources). We also have an effort to map the community at large—who they are, where they live, what they do and whom they work with. olpcMAP.net maps out deployments, projects and volunteers from Armenia to Zambia, with language

translation efforts from Amazigh to Zulu.

Kids Like Sugar

Looking past the hardware, one of the key features of OLPC deployments is its software—more precisely, its software as a learning platform. Sugar is a learning platform designed for children. It originally was designed for the OLPC XO laptop, but since then it has grown to run on most Netbooks and PCs. Sugar is used by more than two million students in Peru, Uruguay, Rwanda, Nepal, the United States and more than 40 other countries.

Sugar is distributed as free software, released under GPL v2. It is available in more than 25 languages, although the translation infrastructure has 132 languages in various stages of completion. Sugar is a collaborative learning platform that promotes collaborative learning. Sugar also can be viewed as a graphical user interface and a collection of software applications (called Activities) that give children a rich environment for learning.

The Sugar user interface is very simple. It has a “low floor”, so that even children as young as ages two and three can start to use it. At the same time, it puts no limit on what they can achieve. Children can use Sugar to reach very complex ideas. They are not bound by its simplicity; rather, they use its simplicity as a catalyst for growth. Sugar presents no “ceiling” to the learner. In fact, each application ships with its source code (usually Python) and can be viewed and modified in much the same way as with HTML.

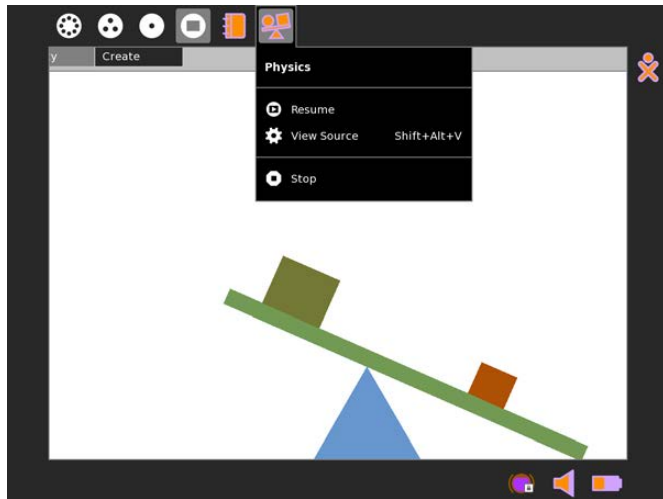


Figure 3. Sugar View Source Option

Sugar has built-in mechanisms for collaboration using an XMPP-based layer that allows for applications to share data across the network. For instance, the Write Activity (based on AbiWord) can “talk” to another child’s Write Activity over the network and allow for real-time, collaborative word processing. This is very much like what you may see with Google Docs, but without Google in the middle. It’s agnostic of the network layer, so the collaboration can work between two laptops meshed together under a tree or two laptops connected over the Internet reaching across continents.

Sugar maintains a journal, a diary of sorts, of everything a child does. It is a record of both what things learners make and how they made them. This journal runs on top of an XML datastore written in Python and works as a flat, time-based listing, sortable by time, topic, mime-type and so on, and it allows children to resume work where they left off. The journal also allows for reflecting upon one’s work by taking notes once each activity

is done. Looking at these notes over time should give students a sense of how their work has evolved. In all, Sugar is not about instruction; it is about learning.

Sugar Activities come in many flavors. Some, such as the Browse activity (based on Firefox) rely on the network extensively, while others, such as the Wikipedia bundle, are completely off-line. Then, there’s the physical computing kind of activity. Activities like Scratch and Turtle Art interface with the 3-D real world using Arduino boards and LEGO robots. Taking a cue from the Uruguayans who are building Arduino-based XO robots and the Peruvians who are distributing 92,000 LEGO robotics kits to their schools, we have started exploring the use of LEGO WeDo robotics kits in Jamaica. The charge is being led by a ten-year-old Jamaican kid!

Then there is the Measure activity, an oscilloscope of sorts, reading from the microphone port for sensor data. How many overtones do throat singers of Tuva produce? Measure can tell you—think lemon batteries, conductive playdough and electromagnetic

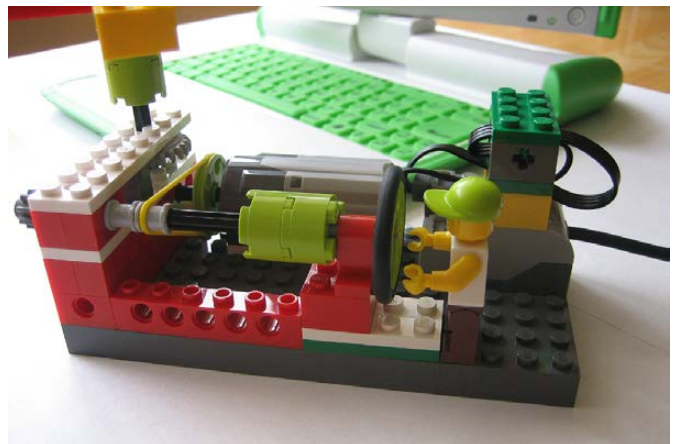


Figure 4. Scratch and LEGO WeDo Robot

speedometers. Now we are talking real education! The list goes on, but you can explore all of these on the Sugar Activities repository.

On the technical side, all of OLPC's XO images are Fedora-based (Fedora 14 at the time of this writing). Sugar runs on top of Fedora (OLPC also packs a GNOME option for older kids). Sugar is primarily written in Python, using PyGTK extensively. Sugar uses an interesting security spec called Bitfrost. Creating Activities in Sugar is easy once you understand Sugar's human interface guidelines. Many existing GTK apps can be "sugarized" easily as well. Although those details are out of the

scope of this article, a good place to start is James Simmons' book *Make Your Own Sugar Activities!* (see Resources).

Sugar Labs is the current home of the Sugar Project. Started as a spin-off from OLPC, it is headed by Walter Bender as its Executive Director. Sugar Labs is a volunteer-driven, nonprofit organization, and a member project of the Software Freedom Conservancy. It coordinates volunteers—an international community of teachers, software developers, artists, writers, parents and children—who are passionate about providing educational opportunities to children through the Sugar Learning Platform. Children as young as 12 years old have written Sugar Activities. When

Powerful: Rhino



Rhino M6500/E6510

- Dell Precision M6500 w/ Core i7 Quad (8 core)
- Dell Latitude E6510 w/ 2.53-2.8 GHz Core i5/i7
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3800M
- 250-750 GB hard drive
- Up to 32 GB RAM (1333 MHz)
- DVD±RW or Blu-ray
- 802.11a/b/g/n
- Starts at \$1385

- High performance NVidia 3-D on a WUXGA RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven



Raven X201 Tablet

- ThinkPad X201 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 2.0-2.13 GHz Core i7
- Up to 8 GB RAM
- 250-500 GB hard drive / 160 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at \$1940

Rugged: Tarantula



Tarantula CF-31

- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA TouchScreen
- 2.4-2.53 GHz Core i5
- Up to 8 GB RAM
- 160-750 GB hard drive / 256 GB SSD
- Call for quote

EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

1-888-651-6686



a 12-year-old developer walked into one of our OLPC San Francisco meetings, we thought he was lost and looking for his parents!

Conclusion

In all, the project is alive and well. Every single laptop out there counts as one more platform of information—both for access and for creation. Young minds all over are starting to get opportunities to connect the dots with the rest of the world. These are not cheap laptops. These are little Linux-powered revolution machines, and the revolution is on. So, come and join us. Bring meaningful and participatory education to millions. Stand up and make your mark for those who cannot. Help them solve their own problems. Start tiny sparks, so that some day, we may light up the world.

Before I wrap up, I should set the record straight. I have been approached by several people who ask, “Didn’t OLPC switch over to Windows?” This myth also is a popular point of contention on sites like Slashdot to which John Walington (OLPC’s VP for Hardware Engineering) replied: “Linux has shipped, and will continue to ship, on every XO produced. You can believe random commenters, or you can believe the person who signs off on every SKU produced.” Two million+ laptops *all* shipped with Linux. There you have it, straight from the man.

Acknowledgements

I would like to acknowledge the input and suggestions from Walter Bender, Christoph Derndorfer, Adam Holt and Mike Lee. ■

Sameer Verma is a professor of Information Systems in the College of Business at San Francisco State University in San Francisco, California. He is also the founder and chief organizer of the OLPC San Francisco volunteer community. All of his computers run Linux—even the one in his car. Track him down at <http://verma.sfsu.edu>.

Resources

One Laptop Per Child: <http://laptop.org>

Sugar Labs: <http://sugarlabs.org>

OLPC Planet: <http://planet.laptop.org>

Sugar Labs Planet: <http://planet.sugarlabs.org>

Sugar Translation: <http://translate.sugarlabs.org>

Sugar Activities Repository:
<http://activities.sugarlabs.org>

OLPC San Francisco: <http://olpcsf.org>

Bhagmalpur Project:
<http://bhagmalpur.wordpress.com>

Jamaica Projects: <http://olpcjamaica.org.jm>

Tuva Project: <http://olpctuva.wordpress.com>

Scratch and LEGO WeDo:
<http://info.scratch.mit.edu/WeDo>

James Simmons’ *Make Your Own Sugar Activities!*: <http://en.flossmanuals.net/make-your-own-sugar-activities>

John Walington’s Comment on Slashdot:
<http://hardware.slashdot.org/comments.pl?sid=1941464&cid=34810832>